Signal Processing: *Image Communication* 27 (2012) 96–111

Contents lists available at ScienceDirect

Signal Processing: *Image Communication*

journal homepage: www.elsevier.com/locate/image

Human action recognition using Pose-based discriminant embedding

Behrouz Saghafi, Deepu Rajan *

Centre for Multimedia and Network Technology, School of Computer Engineering, Nanyang Technological University, Singapore 639798, Singapore

ARTICLE INFO

Article history:
Received 19 November 2010
Accepted 9 May 2011
Available online 27 May 2011

Keywords:
Human action recognition
Dimensionality reduction
Discriminant embedding
Silhouette
Posture
Action period

ABSTRACT

Manifold learning is an efficient approach for recognizing human actions. Most of the previous embedding methods are learned based on the distances between frames as data points. Thus they may be efficient in the frame recognition framework, but they will not guarantee to give optimum results when sequences are to be classified as in the case of action recognition in which temporal constraints convey important information. In the sequence recognition framework, sequences are compared based on the distances defined between sets of points. Among them Spatio-temporal Correlation Distance (SCD) is an efficient measure for comparing ordered sequences. In this paper we propose a novel embedding which is optimum in the sequence recognition framework based on SCD as the distance measure. Specifically, the proposed embedding minimizes the sum of the distances between intra-class sequences while seeking to maximize the sum of distances between inter-class points. Action sequences are represented by key poses chosen equidistantly from one action period. The action period is computed by a modified correlation-based method. Action recognition is achieved by comparing the projected sequences in the low-dimensional subspace using SCD or Hausdorff distance in a nearest neighbor framework. Several experiments are carried out on three popular datasets. The method is shown not only to classify the actions efficiently obtaining results comparable to the state of the art on all datasets, but also to be robust to additive noise and tolerant to occlusion, deformation and change in view point. Moreover, the method outperforms other classical dimension reduction techniques and performs faster by choosing less number of postures.

1. Introduction

Recognition of human actions from video is an important problem in computer vision that can take advantage of signal processing techniques. It has applications in a wide variety of topics including content-based video retrieval, human–computer interaction and surveillance. The problem is challenging especially in cases of intra-class variations in appearance and of different actions with similar postures.

Some of the common features used in action recognition are optical flow [1], space-time gradients [2], point trajectories [3] and sparse interest points [4–6]. Optical flow vectors are often inaccurate when the videos are of low quality and especially when motion is not smooth. Moreover, it is not robust to illumination changes. Likewise, point trajectories need accurate tracking methods which could fail in cases of fast moving subjects, occlusions and cluttered backgrounds. Also by using sparse representation of interest points, we lose the global structural information, which could otherwise help in the recognition process. On the other hand, silhouettes are informative features for describing actions [7–13].

* Corresponding author.
E-mail address: asdrajan@ntu.edu.sg (D. Rajan).

0923-5965/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.image.2011.05.002

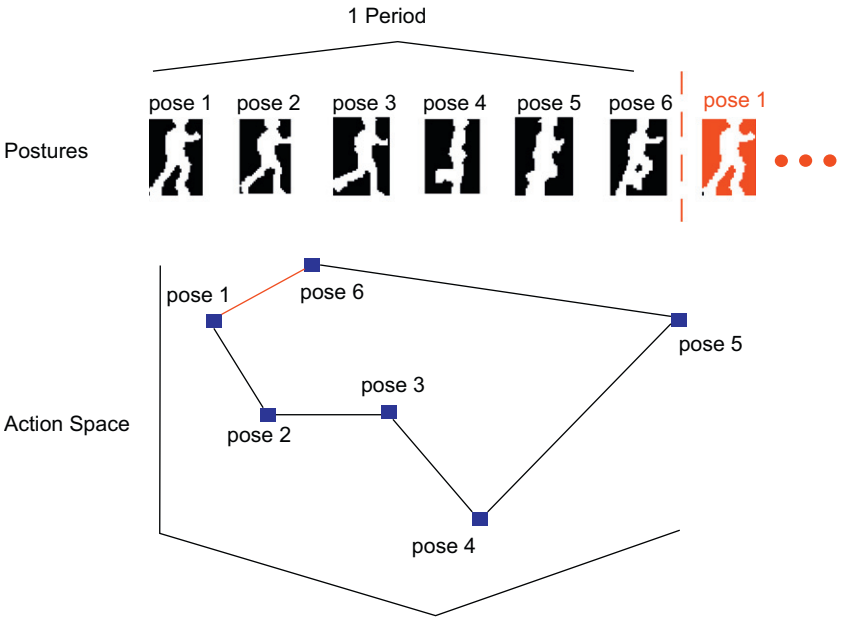

Fig. 1. Examples of postures for *run* and its trajectory in a possible action space.

They are able to capture the spatio-temporal characteristics of motion with possibly lower computational costs [7]. There is no need for an explicit model of the human body. Furthermore, recent advances in foreground extraction from complex backgrounds and in the presence of camera motion have benefited from improved models for segmentation and global motion extraction. In particular, segmentation algorithms have benefited from the area of alpha matting in which a pixel is composed of both background and foreground components and the problem is to estimate these components [14–16].

There have been two general frameworks for using silhouettes in action recognition. Some approaches classify action sequences on a frame-by-frame basis [8]. Thus each frame is independently classified as belonging to one of the actions. The label for the query sequence is obtained based on a voting scheme. All these approaches belong to the *frame recognition framework*. These methods ignore the temporal information and kinematics which is useful in the classification. Then, there are methods that classify the sequence as a whole [9]. These approaches belong to the *sequence recognition framework*. These methods compare sequences based on distances like Spatio-temporal Correlation Distance (SCD) or Hausdorff distance, which are defined between sequences of points. Human action, when represented as a sequence of silhouettes, can be considered as a function of time in which the silhouette of the body changes gradually. Thus, motion information is also included in this kind of representation without using expensive motion features that are difficult to extract. In this paper we utilize the latter framework.

Silhouettes can be considered as points in high-dimensional image space. Consequently, action sequences are described as data trajectories inside image space. Recognition methods which operate in this high-dimensional space suffer from the curse of dimensionality. In addition, the information provided in the high-dimensional image space is way more than required to describe an action. Moreover, the structure of the human body imposes a constraint on possible postures. Hence, it is more efficient to analyze action trajectories in a lower dimensional space. We call this subspace *the action space*. Examples of postures from the action *run* as well as the trajectory in a possible action space is shown in Fig. 1.

1.1. Motivation and overview of approach

There have been previous efforts at learning an efficient action space in order to classify actions. Accordingly, general dimension reduction techniques such as Principal Components Analysis (PCA), Linear Discriminant Analysis (LDA), Locality Preserving Projections (LPP) [9], Locally Linear Embedding (LLE) [17], Laplacian Eigenmaps (LE) [18] and Kernel PCA [10], as well as action-specific embeddings like Local Spatio-Temporal Discriminant Embedding (LSTDE) [8] have been used. These methods are explained in more details in the next section. In all these methods, the embedding is defined based on the distance between data points rather than the distance between sequences; thus, they may be efficient in the *frame recognition framework*, but they are not guaranteed to give optimum results when *sequences* are classified. As stated earlier, in the sequence recognition framework, the query sequence is compared to the learned sequences based on distances generally defined between sets of data points, like SCD. In this paper we develop a novel embedding which is optimum in the sequence recognition framework. Specifically, the proposed embedding minimizes sum of the distances between intra-class sequences while maximizing the total distances between inter-class points.

SCD is an effective distance between ordered sequences. In order to compute SCD between two sequences, they need to have the same lengths. First, two sequences are shifted

Fig. 2. Training phase of the proposed method.

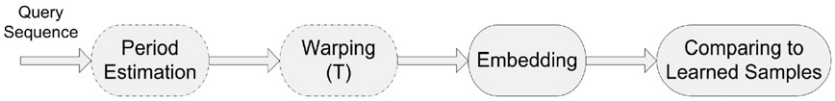

Fig. 3. Testing phase of the proposed method.

circularly to ensure that the frames corresponding to similar poses are aligned together. Then the distance between corresponding frames is calculated and summed to compute the overall distance. In order to find the optimum embedding based on SCD, we represent sequences by a fixed number of postures chosen equidistantly in one action period and obtained by interpolation. The proposed embedding is such that in the action space, same postures from same actions (executed by different individuals) are embedded close, while postures from different actions are embedded as far apart as possible. Since this embedding acts on each pose of the action, we call it Pose-based Discriminant Embedding (PDE).

Fig. 2 shows the flowchart for the training phase of the proposed method. First the action period in the training sequences is estimated based on a modification of the method of Cutler et al. [19]. For computational efficiency only one period of each sequence is used. Sequences are then warped to have the same length. Subsequently, for intra-class sequences, similar postures are aligned together. Eventually the PDE projection matrix is computed. Details of each part are explained in the next sections.

The block diagram for the testing phase is illustrated in Fig. 3. During test, period estimation and warping are needed only when SCD is used as the distance metric and hence, they are shown in dotted blocks. They are not required when using the Hausdorff distance, although the accuracy of recognition is increased when they are used. The query sequence is embedded into the action space based on the embedding matrix computed in the training phase and compared to the trained sequences in the low-dimensional action space using SCD or Hausdorff distance. A simple nearest neighbor classifier is used to classify the sequence.

Our proposed method guarantees to give the most discriminant action space in order to be used with SCD. Since the sequence-based recognition framework is more efficient than frame-based, the proposed method outperforms other dimension reduction methods like PCA, LDA and LPP in action recognition. Also our method performs faster and is more computationally efficient compared to the aforementioned embeddings by using less number of frames (postures). Moreover, our method is robust to additive noise, occlusion and deformation. Also viewpoint change is tolerated up to a considerable extent.

1.2. Organization

The paper is organized as follows. Section 2 reviews some related works on dimension reduction techniques used for action recognition. The distance metrics used between sequences are described in Section 3. Subsequently the proposed embedding method as well as the objective functions used for optimization is detailed. Section 4 describes the steps required for preprocessing. Experimental results are presented and discussed in Section 5. Finally we conclude the paper in Section 6.

2. Related work

The corpus of literature in action recognition is expanding rapidly. Review articles on action recognition can be found in [20,21]. It is not the intention here to present an exhaustive survey of the various methods for action recognition. Instead, we focus especially on dimension reduction methods used in action recognition. While many dimension reduction techniques have been used successfully for subspace learning for face recognition, only few dimension reduction methods have been used for learning the action space for action recognition. Three of the common dimension reduction techniques including PCA, LDA and LPP have been used by Wang and Suter [9] to discover the underlying action manifold. PCA simply chooses the directions in which the data has maximum variance [22]. It is intrinsically an unsupervised technique. LDA is a supervised method, which tries to make data discriminative for better classification [23]. This means that in the embedded space, intra-class data points are as close as possible while inter-class points are as far apart as possible. Although it is generally believed that LDA performs better than PCA, yet when the training set is small, PCA can outperform LDA [24]. LPP tries to preserve the distance between data points in the target space so that local structure is preserved [25]. It is used in either supervised or unsupervised manner.

All the methods mentioned so far are linear. LE [18], LLE [17] and Kernel PCA [10] are nonlinear techniques which have been used for action recognition. Experiments performed by Wang and Suter [9] have verified that linear methods (PCA, LDA and LPP) outperform the nonlinear ones (LE and LLE) in action recognition. The reason for

this is the complexity in parameter adjustment and extrapolation in these nonlinear methods [9]. Another disadvantage of nonlinear dimension reduction techniques is that they are computationally expensive.

The concept of Canonical Correlation Analysis (CCA) or Principal Angles [26] has also been used for manifold learning which models the set of intra-class frames as a hyper plane. Similar to optimization concept in LDA, Kim et al. [27] have developed a discriminant analysis using CCA. In this framework which is used to classify the image sets, the canonical correlations of inter-class sets are maximized, while minimizing the canonical correlations related to intra-class sets. The optimal embedding is computed in an iterative learning manner. In a similar work, Wu et al. [11] have proposed an online framework, which incrementally update the discriminative model upon adding online training samples using the eigenspace merging algorithm. These methods ignore the temporal constraints in action sequences which may be useful in classification. Jia and Yeung [8] have developed another embedding which is discriminative in two folds: spatial and temporal. Their criteria for spatial discrimination are similar to LDA. However, for temporal discrimination, they find the embedding such that the Principal Angles between inter-class temporal subspaces are maximized. The temporal subspace is formed by a short video segment around each frame. Their method does not create a dichotomy between spatial and temporal analysis since the action is represented as a sequence of silhouettes that contain spatio-temporal changes.

As an instance for considering temporal constraints, Nayak et al. [28] have proposed learning a subspace of distributions for recognition of articulated activities. The action is represented as frame-wise distributions of low level features such as orientation, color, or relational distributions. As time grows, the configuration of parts changes which result in changing the distribution in the latent space. So the whole action will result in a trajectory, which can be compared in the latent space. The experiments have been done on gesture recognition and classification of human–human interaction sequences.

While most of the corpus of manifold learning approaches are based on representing images or frames as pixel-wise vectors, Torki and Elgammal [29] have proposed to learn manifolds from a collection of local features such that it captures the feature similarities and spatial structures. By choosing proper affinity metrics between feature descriptors and spatial coordinates, first the training set is embedded into a new space. Subsequently features of the new image are embedded using a coordinate propagation method. This method has been successfully verified on shape and object recognition datasets.

3. Pose-based discriminant embedding

In the sequence recognition framework, sequences of silhouettes are embedded into a lower dimensional space called action space. The sequences are compared in the action space using distances defined between sets of data points.

Each frame with the resolution of $M \times N$ is converted into a vector $\boldsymbol{f}$ of dimension $h = M \times N$ in the lexicographic order. Let $\boldsymbol{f}_i(t)$ represent the frame at the time t from the ith sequence. Then the ith sequence, $\boldsymbol{F}_i$ can be considered as a function of time by $\boldsymbol{F}_i = \boldsymbol{f}_i(t), t = 1, \ldots, N_i$, where N_i is the number of frames in $\boldsymbol{F}_i$. The total number of training samples for n training sequences is $N_t = N_1 + \cdots + N_n$. The overall training set is denoted by

$$\begin{aligned}\boldsymbol{X} &= [\boldsymbol{f}_1(1)\boldsymbol{f}_1(2), \ldots \boldsymbol{f}_1(N_1)\boldsymbol{f}_2(1), \ldots \boldsymbol{f}_n(N_n)] \\ &= [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{N_t}]. \end{aligned} \tag{1}$$

Thus, $\boldsymbol{X}$ is a matrix of size $h \times N_t$. Each data point $\boldsymbol{x}_i$ in the h-dimensional image space is embedded into a point $\boldsymbol{y}_i$ in the l-dimensional ($l \ll h$) action space by $\boldsymbol{y}_i = \boldsymbol{A}^T \boldsymbol{x}_i$, where $\boldsymbol{A}$ is the embedding matrix. Specifically, the sequence $\boldsymbol{F}_i = \boldsymbol{f}_i(t), t = 1, \ldots, N_i$ is embedded into sequence $\boldsymbol{Q}_i = \boldsymbol{q}_i(t), t = 1, \ldots, N_i$ in the action space. Embedded sequences are compared based on the distances defined between sets of points. We first review these distances and then describe how the proposed embedding is computed.

3.1. Distances between sets of points

3.1.1. Median Hausdorff distance

The Hausdorff distance measures the similarity of two data point sets, by finding the points in one set which are close to points in the other set and vice versa. The MHD from a sequence $\boldsymbol{Q}_1$ to a sequence $\boldsymbol{Q}_2$ is defined as [9]:

$$d_{MHD}(\boldsymbol{Q}_1, \boldsymbol{Q}_2) = \text{median}_i \left(\min_j (\|\boldsymbol{q}_1(i) - \boldsymbol{q}_2(j)\|) \right), \tag{2}$$

where i and j refer to time instances. We believe that *Median Hausdorff Distance* is a better choice than *Mean Hausdorff Distance* since the former is not affected by outliers. To incorporate symmetry, the final distance measure used is

$$D_{MHD}(\boldsymbol{Q}_1, \boldsymbol{Q}_2) = d_{MHD}(\boldsymbol{Q}_1, \boldsymbol{Q}_2) + d_{MHD}(\boldsymbol{Q}_2, \boldsymbol{Q}_1). \tag{3}$$

When using MHD, two sequences do not need to be of the same length. Moreover, MHD ignores the order of frames in the sequences.

3.1.2. Spatio-temporal correlation distance

SCD considers the arrangement of frames in sequences; thus, it is more efficient for sequence recognition and particularly action recognition. The two sequences should have the same length in order to be compared. SCD between sequences $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$ is defined as [9]:

$$D_{SCD}(\boldsymbol{Q}_1, \boldsymbol{Q}_2) = \min_b \sum_{t=1}^{T} \|\boldsymbol{q}_1'(t) - \boldsymbol{q}_2'(t+b)\|^2, \tag{4}$$

where $\boldsymbol{Q}_1' = \boldsymbol{q}_1'(t)$ and $\boldsymbol{Q}_2' = \boldsymbol{q}_2'(t)$; $t = 1, \ldots, T$ are warped versions of the sequences $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$. In Section 4, we explain how to warp the sequences into the same length. The variable b stands for circular time shifting to align the corresponding frames together for comparison. By aligning we ensure that Eq. (4) gives the minimum distance among the different alignments. SCD is basically the sum of Euclidean distances between corresponding points that represent frames in the embedded space.

As explained earlier, MHD ignores the order of points in the sequences. However, the order of silhouettes in the sequences can be useful in the classification process. Thus in order to find the optimal embedding we use SCD, which is based on ordered sequences. Note that SCD needs the action period to be computed. Since estimating the action period is not always easy in real videos, in this paper we also use MHD in order to compare the sequences.

3.2. Optimal embedding computation

In this paper we propose an embedding such that in the embedded space (action space), the sequences are as discriminant as possible using SCD as the distance metric. Let $\boldsymbol{A}$ denote the embedding matrix. In order for intra-class sequences to be as close as possible in the action space, the sum of all pairwise SCD between embedded intra-class sequences should be minimized with respect to $\boldsymbol{A}$:

$$\min_{\boldsymbol{A}} \sum_{c} \sum_{i,j \in C_c} D_{SCD}(\boldsymbol{A}^T \boldsymbol{F}_i, \boldsymbol{A}^T \boldsymbol{F}_j) = \min_{\boldsymbol{A}} \sum_{c} \sum_{i,j \in C_c} \sum_{t=1}^{T} \|\boldsymbol{A}^T \boldsymbol{F}_i(t) - \boldsymbol{A}^T \boldsymbol{F}_j(t)\|^2; \quad (5)$$

where C_c represents the set of all sequences belonging to class c. During preprocessing, the sequences are warped to the same length T and the intra-class sequences are aligned by applying the circular time shifting in Eq. (4). In order to rewrite Eq. (5) in terms of $\boldsymbol{x}$, we define the $N_t \times N_t$ matrix $\boldsymbol{W}_p$ such that

$$\boldsymbol{W}_{p_{ij}} = \begin{cases} 1 & \text{if } \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ are similar postures from the same action,} \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

This way the data points $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{N_t}\}$ are modeled as the nodes of a graph G_p with $\boldsymbol{W}_p$ as its affinity matrix. Therefore Eq. (5) is converted to

$$\min_{\boldsymbol{A}} \sum_{ij} \|\boldsymbol{A}^T \boldsymbol{x}_i - \boldsymbol{A}^T \boldsymbol{x}_j\|^2 \boldsymbol{W}_{p_{ij}}. \quad (7)$$

For ease of derivation, we rewrite the objective function of (7) in the *trace* format as

$$\begin{aligned} \frac{1}{2} \sum_{ij} \|\boldsymbol{A}^T \boldsymbol{x}_i - \boldsymbol{A}^T \boldsymbol{x}_j\|^2 \boldsymbol{W}_{p_{ij}} &= \frac{1}{2} \sum_{ij} \mathrm{Tr}\{\boldsymbol{A}^T (\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \boldsymbol{A}\} \boldsymbol{W}_{p_{ij}} \\ &= \frac{1}{2} \mathrm{Tr}\{\boldsymbol{A}^T \sum_{ij} ((\boldsymbol{x}_i - \boldsymbol{x}_j) \boldsymbol{W}_{p_{ij}} (\boldsymbol{x}_i - \boldsymbol{x}_j)^T) \boldsymbol{A}\} \\ &= \mathrm{Tr}\{\boldsymbol{A}^T (\boldsymbol{X} \boldsymbol{D}_p \boldsymbol{X}^T - \boldsymbol{X} \boldsymbol{W}_p \boldsymbol{X}^T) \boldsymbol{A}\} \\ &= \mathrm{Tr}\{\boldsymbol{A}^T \boldsymbol{X} \boldsymbol{L}_p \boldsymbol{X}^T \boldsymbol{A}\}, \end{aligned} \quad (8)$$

where $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{N_t}]$, $\boldsymbol{D}_p$ is a diagonal matrix with column (or row) sums of symmetric $\boldsymbol{W}_p$ as entries, and $\boldsymbol{L}_p = \boldsymbol{D}_p - \boldsymbol{W}_p$ forms the Laplacian matrix of the graph G_p. Since each entry of $\boldsymbol{D}_p$ is a row (or column) sum of $\boldsymbol{W}_p$, a large value of $\boldsymbol{D}_p(i,i)$ indicates the higher importance of the associated point. Therefore, the following constraint is imposed:

$$\mathrm{Tr}\{\boldsymbol{A}^T \boldsymbol{X} \boldsymbol{D}_p \boldsymbol{X}^T \boldsymbol{A}\} = 1, \quad (9)$$

so that the objective function in Eq. (8) turns into

$$\min_{\boldsymbol{A}} \{1 - \mathrm{Tr}\{\boldsymbol{A}^T \boldsymbol{X} \boldsymbol{W}_p \boldsymbol{X}^T \boldsymbol{A}\}\} \quad (10)$$

or

$$\max_{\boldsymbol{A}} \mathrm{Tr}\{\boldsymbol{A}^T \boldsymbol{X} \boldsymbol{W}_p \boldsymbol{X}^T \boldsymbol{A}\}. \quad (11)$$

In order for inter-class sequences to be as far apart as possible based on SCD, we need to align each pair of inter-class sequences when compared together. But sequences are already aligned based on intra-class labels. So the alignment cannot be done for inter-class labels. For instance, suppose we have sequences $\boldsymbol{F}_1, \boldsymbol{F}_2, \boldsymbol{F}_3$ and $\boldsymbol{F}_4$ for which $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$ belong to class C_1 and $\boldsymbol{F}_3$ and $\boldsymbol{F}_4$ belong to class C_2. $\boldsymbol{F}_2$ is aligned based on $\boldsymbol{F}_1$ and similarly $\boldsymbol{F}_4$ is aligned based on $\boldsymbol{F}_3$. When comparing $\boldsymbol{F}_1$ and $\boldsymbol{F}_4$, $\boldsymbol{F}_4$ cannot be aligned based on $\boldsymbol{F}_1$, because it has already been aligned based on $\boldsymbol{F}_3$. Thus for optimization of inter-class sequences, we consider the distance between data points rather than sequences as for the previous discriminative embeddings. So we define the affinity matrix $\boldsymbol{W}_c$ such that

$$\boldsymbol{W}_{c_{ij}} = \begin{cases} 1 & \text{if } \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ belong to different actions,} \\ 0 & \text{otherwise.} \end{cases} \quad (12)$$

Consequently the objective function which has to be maximized is

$$\max_{\boldsymbol{A}} \sum_{ij} \|\boldsymbol{A}^T \boldsymbol{x}_i - \boldsymbol{A}^T \boldsymbol{x}_j\|^2 \boldsymbol{W}_{c_{ij}}. \quad (13)$$

Similar to Eq. (8), the objective function of (13) can be written in trace form as

$$\max_{\boldsymbol{A}} \mathrm{Tr}\{\boldsymbol{A}^T \boldsymbol{X} \boldsymbol{L}_c \boldsymbol{X}^T \boldsymbol{A}\}, \quad (14)$$

where $\boldsymbol{L}_c = \boldsymbol{D}_c - \boldsymbol{W}_c$ ($\boldsymbol{D}_c$ is the diagonal matrix with column (or row) sums of $\boldsymbol{W}_c$ as entries). Given the objective functions in Eqs. (14) and (11), together with the constraint in Eq. (9), we have the overall optimization problem for PDE as

$$\max_{\boldsymbol{A}} \{\mathrm{Tr}\{\boldsymbol{A}^T \boldsymbol{X} \boldsymbol{W}_p \boldsymbol{X}^T \boldsymbol{A}\} + \mathrm{Tr}\{\boldsymbol{A}^T \boldsymbol{X} \boldsymbol{L}_c \boldsymbol{X}^T \boldsymbol{A}\}\} \text{s.t.} \mathrm{Tr}\{\boldsymbol{A}^T \boldsymbol{X} \boldsymbol{D}_p \boldsymbol{X}^T \boldsymbol{A}\} = 1. \quad (15)$$

An optimal $\boldsymbol{A}$ can be obtained by finding the l largest eigenvalues of the following generalized eigenvalue problem:

$$(\boldsymbol{X}(\boldsymbol{W}_p + \boldsymbol{L}_c)\boldsymbol{X}^T)\boldsymbol{a} = \lambda(\boldsymbol{X} \boldsymbol{D}_p \boldsymbol{X}^T)\boldsymbol{a}. \quad (16)$$

The proposed embedding method projects sequences into a space in which the intra-class sequences are close together (in terms of SCD) and also the inter-class frames are as far apart as possible. While the latter is similar to LDA, the former characteristic enables our method to clearly outperform LDA and other similar discriminant embeddings in the sequence recognition framework. LDA tries to minimize the distance between intra-class points, but may not necessarily give the least SCD possible between intra-class sequences, which our method seeks. When comparing the sequences based on SCD it will have less classification error compared to LDA. This superiority becomes even more obvious in the cases of similar inter-class actions like run,

walk and skip as well as when the silhouettes are noisy. In these situations inter-class sequences may be originally closer than intra-class ones. PDE enforces the intra-class sequences to be closer in the action space compared to LDA leading to less error in classification.

The other characteristic of the proposed embedding is that in the action space similar postures from the same actions (performed by different subjects) are embedded as close as possible. Fig. 4 shows two trajectories of *run* performed by two different persons. The change in appearance has resulted in deviation of postures in the action space. The embedding aims to remove these noise-like deviations from trajectories and make them as close as possible so as to minimize the classification error. In other words, it aims to make all intra-class trajectories performed by different subjects, coincide in the action space.

4. Preprocessing

As shown in Fig. 2 we perform some preprocessing on the training sequences before computing the PDE. Preprocessing involves period estimation, warping and aligning. The first two steps reduce the computational complexity as well as time by choosing less number of frames. Similar postures from intra-class actions are also aligned for embedding computation. Moreover, as illustrated in Fig. 3 the query sequences are also preprocessed for period estimation and warping.

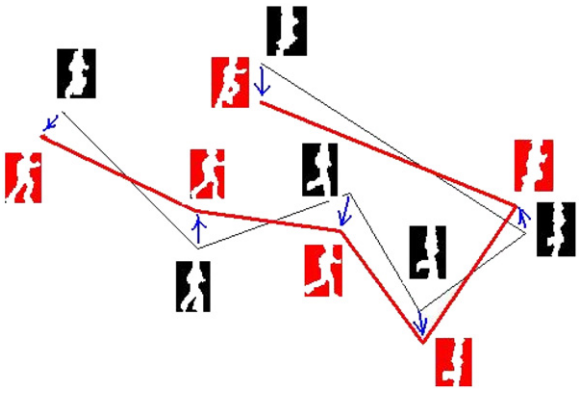

Fig. 4. Trajectories of one period of action run with the same duration for two different persons (red and black). Arrows show the deviations due to appearance change. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.1. Period estimation

Action sequences can be considered as periodic repetitions of an action cycle. Using a single period is much more computationally efficient than using the entire length of the video sequence. For computing the action period, we use the method of Cutler and Davis [19], which is also used in [9]. However, we modify it slightly for more accurate estimation. The distance between frames at times t_1 and t_2 is computed as [19]:

$$\boldsymbol{S}_{t_1,t_2} = \sum_{(x,y)} |\boldsymbol{B}_{t_2}(x,y) - \boldsymbol{B}_{t_1}(x,y)|, \tag{17}$$

where $\boldsymbol{B}_{t_1}$ and $\boldsymbol{B}_{t_2}$ are the silhouettes in times t_1 and t_2, respectively, which have been centered and normalized into the same dimensions. $\boldsymbol{S}$ will have a periodic pattern as shown in Fig. 5(a), in which darker regions indicate less distance. For periodic action sequences (like walk in Weizmann database), dark lines are arranged parallel to the diagonal of $\boldsymbol{S}$. To determine the action period, one arbitrary column vector $\boldsymbol{z}$ of $\boldsymbol{S}$ is chosen and linearly detrended to obtain the new vector $\hat{\boldsymbol{z}}$ (Fig. 5(b)). Then the autocorrelation of $\hat{\boldsymbol{z}}$ is computed (Fig. 5(c)). In [9] the action period is estimated as the mean distance between each pair of consecutive peaks in the aforementioned autocorrelation function. Here we use median instead of mean since it leads to more stable results.

We do not use the zero-derivative method to find peak positions as in [9] since a little noise will result in many false detections. Instead, we pick a point as the peak whose left and right neighbors have lower values by some margin [30]. In Fig. 6 the false peak detections which result from applying the zero-derivative method is illustrated, which can be avoided by using our method. For more accurate estimation, the median value of the period estimated for every column of $\boldsymbol{S}$ can be used as the action period.

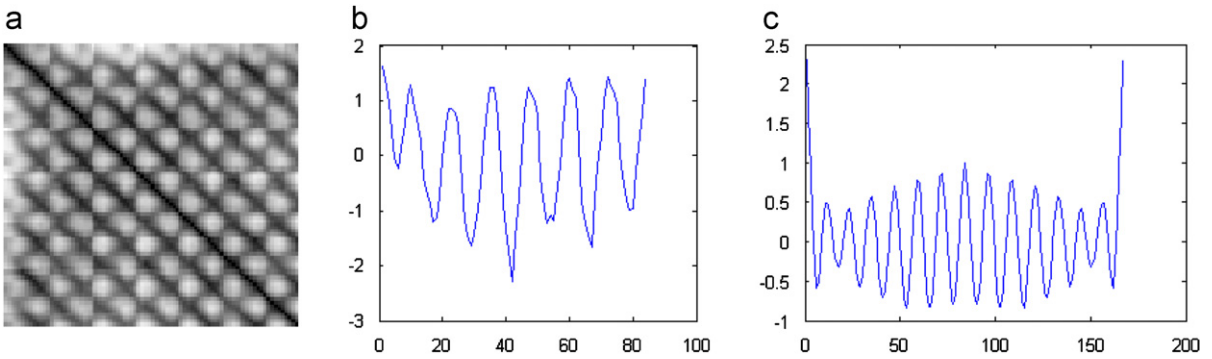

Fig. 5. Period estimation for walk sequence pertaining to person 1 (Daria) of Weizmann database [12] with a length of 84, (a) similarity $\boldsymbol{S}$, (b) $\hat{\boldsymbol{z}}$, the middle column vector $\boldsymbol{z}$ of $\boldsymbol{S}$ which is linearly detrended, (c) its autocorrelation.

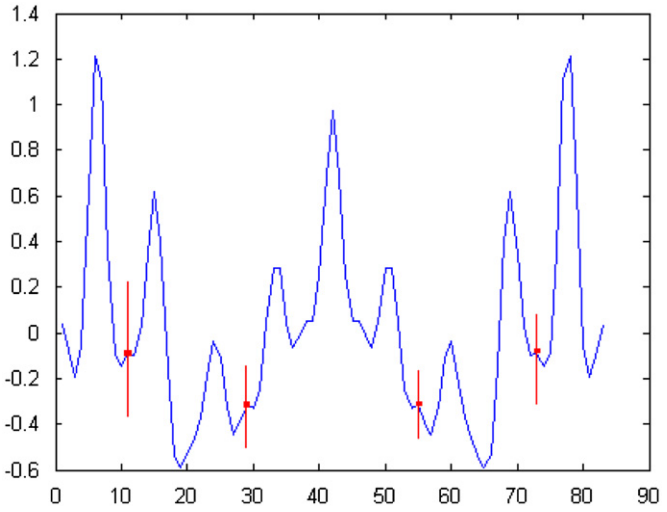

Fig. 6. False peak detections by zero-derivative method indicated by red vertical lines. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

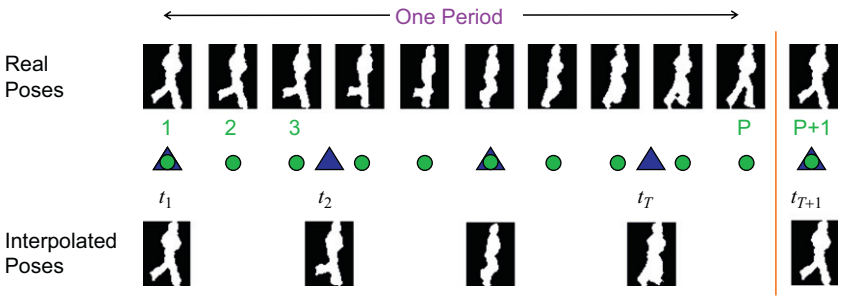

Fig. 7. Finding T interpolation time instances for a P length sequence. On the time axis, the green circle markers show the time instances for existing frames $(1,2,\ldots,P)$ and the blue triangle markers show the interpolation time samples. Real poses from one period of the action *run* are shown above the time axis and the interpolated poses are represented below the axis. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.2. Warping

In order to use SCD, the sequences should have the same length. So we need to warp all the training sequences into the same global length before computing PDE. Warping the test sequence is also necessary when using SCD as the distance metric. However, it is not required for MHD due to the point-set matching properties of this measure.

By warping, we choose less number of frames to be processed. Thus we decrease the computational complexity and time of the processing with the accuracy almost remaining the same. Warping is done on one action period. The action cycle is warped to the length T, by selecting T time instances equidistantly from the period, starting from the beginning, and interpolating frames at the selected times. The interpolation is done using the existing frames by bicubic interpolation technique. Fig. 7 illustrates the warping procedure. The green circle markers show the time instances for existing frames. If the action cycle is P, the posture in time instance $P+1$ is similar to the posture in time 1. The blue triangle markers stand for the time instances of interpolated frames. These time samples are denoted by $t_1,t_2,\ldots,t_T,t_{T+1}$. They are chosen equidistantly during one period. So we have

$$\frac{t_i-t_1}{t_{T+1}-t_1}=\frac{i-1}{(T+1)-1};\quad i=1,2,\ldots,T. \tag{18}$$

Since $t_1=1$ and $t_{T+1}=P+1$, the time instances are chosen by

$$t_i=1+\frac{P}{T}(i-1);\quad i=1,2,\ldots,T. \tag{19}$$

T is chosen less than the minimum possible action period. The effect of T is studied in Section 5.

4.3. Aligning

After warping, we need to align the similar poses of training sequences in the same order before computing the embedding matrix. For this purpose, we employ Eq. (4) used for computing SCD. Here the objective is to compute b, the circular shift and not the distance. For each action in the training set, one sequence is considered as the reference for aligning (e.g. sequence pertaining to person 1). The rest of the intra-class sequences are aligned

with the reference sequence. For example Fig. 8 shows two warped sequences of action run. After aligning, the first pose of the reference sequence is similar to the 5th pose of sequence 2 resulting in $b=4$.

5. Experimental results

We have performed several experiments to show the efficiency and effectiveness of PDE for action recognition. We have used three common action datasets: Weizmann database [12], Maryland database [13] and KTH database [2]. Segmenting the foreground is not the main interest in our work, so we use the silhouette masks which are available in Weizmann and Maryland datasets. For KTH dataset we manually extract silhouettes as described in Section 5.3. All silhouette frames are centered and normalized into the same dimension which is 64×48 for Weizmann and Maryland datasets and 80×48 for KTH dataset.

To avoid the singular matrix problem in the optimization of Eq. (15), we preprocess the data using PCA so that 98% information is kept in the sense of low rank approximation.

5.1. Experiments on Weizmann database

This database contains 10 action classes performed by nine different human subjects. The actions include bending (bend), jumping jack (jack), jumping-forward-on-two-legs (jump), jumping-in-place-on-two-legs (pjump), running (run), galloping sideways (side), skipping (skip), walking (walk), waving-one-hand (wave1), and waving-two-hands (wave2) [12]. In all the experiments, we use the leave-one-out cross-validation method, i.e. each time we leave one sequence out for testing and train with the remaining sequences in the dataset. We report the average of recognition results.

5.1.1. Results and analysis

We warp both the training and test sequences into the temporal duration T. Since the minimum length of the periods is 9 (run), we let T to vary from 3 to 9 in order to study the effect of T. For each value of T, we change the dimension (l) from 1 to 200. SCD is used as the distance metric. The results are shown in Fig. 9. We achieve recognition rates close to 100%, which clearly shows the

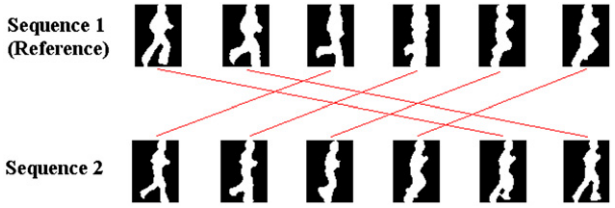

Fig. 8. Aligning two warped sequences of action run: The first pose of sequence 1 is similar to the 5th pose of sequence 2. The other poses are aligned in a cyclic manner. For this example: $b=4$.

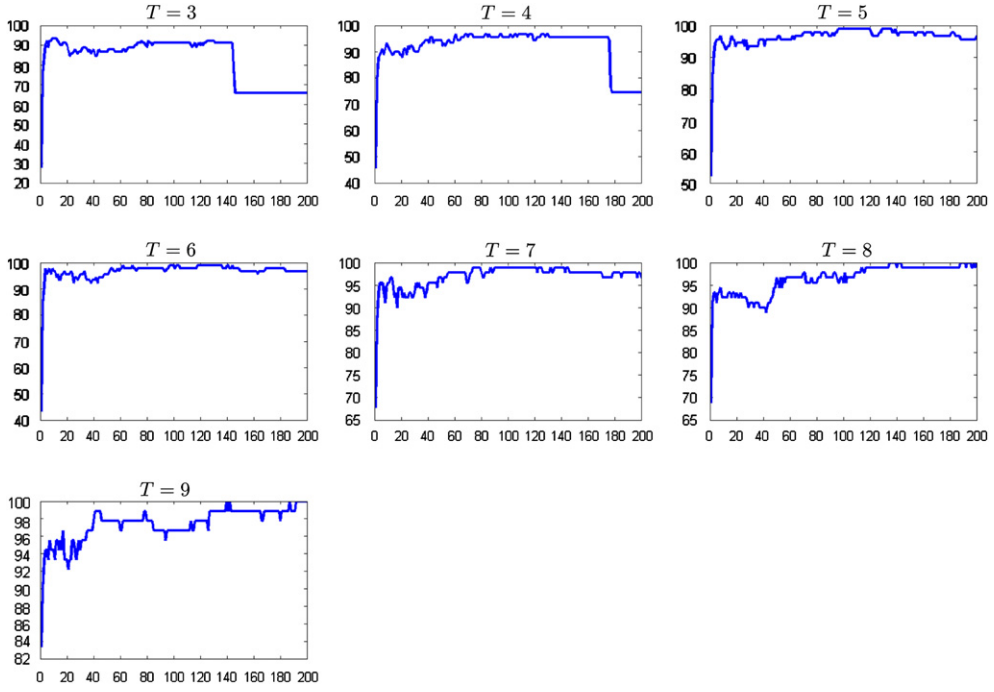

Fig. 9. Recognition accuracy versus dimension (l), for different values of T for SCD as the distance metric.

efficiency of our method in learning the action space. Recognition rate of 100% is achieved for $T=8$ and 9. As seen from the plots in Fig. 9, the accuracy usually does not change considerably with small changes of l. The sudden drop off for $T=3$ and 4 is likely due to *curse of dimensionality*. In other words, by increasing the dimension from this point forward the number of training samples are no longer sufficient to learn the subspace. This occurs for small Ts, since the number of training data points are much lower than for bigger Ts, given the training sequences. The mean training time on a 2.67 GHz CPU ranges from 0.57 s for $T=3$ to 3.09 s for $T=9$, which is considerably low. Also the testing lasts for less than 0.01 s which is significantly fast.

To study the effect of T, we compare the maximum and also mean of recognition accuracy in the stable interval from the plots in Fig. 9 for different values of T. The comparison is illustrated in Fig. 10. As seen from the figure, the accuracy is not so sensitive to the change of T. The best results (100%) occur for $T=8$ and 9. By reducing T, the accuracy almost decreases, since smaller Ts have less number of postures to discriminate between actions.

We also explore using MHD as the distance measure. Similar to Fig. 9 the results using MHD when both the training and test sequences are warped are shown in Fig. 11. Here, too, we achieve recognition rates close to 100%. We obtain 100% recognition accuracy for $T=6$. Similar to SCD, there are drop offs for $T=3$ and 4. The test time using MHD ranges from 0.02 s for $T=3$ to 0.07 s for $T=9$. Testing takes longer compared to using SCD. Since the distance is not used during training phase, the time for training is the same as SCD. Comparing Figs. 9 and 11, the results using MHD seem smoother because SCD is a sum of T distances while Hausdorff distance is simply one of the distance values (the median value).

In order to examine the effect of T, comparison between maximum and average accuracy of different Ts is shown in Fig. 12. For MHD, $T=6$ which is the middle value in our range (3–9) has the best results. Similar to SCD, the accuracy for small Ts is lower. Moreover, for MHD the accuracy for large Ts is slightly lower than $T=6$. This is probably because adjacent frames in a sequence will be similar and they will be no longer discriminant.

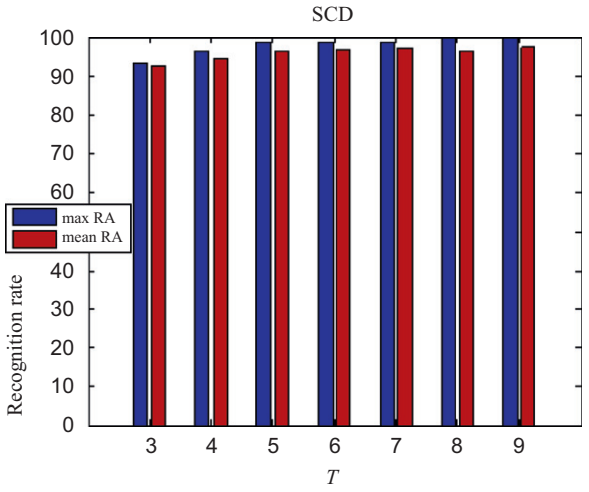

Fig. 10. Studying the effect of T: comparing the maximum and mean of recognition rate in the stable interval from the plots in Fig. 9 for different values of T.

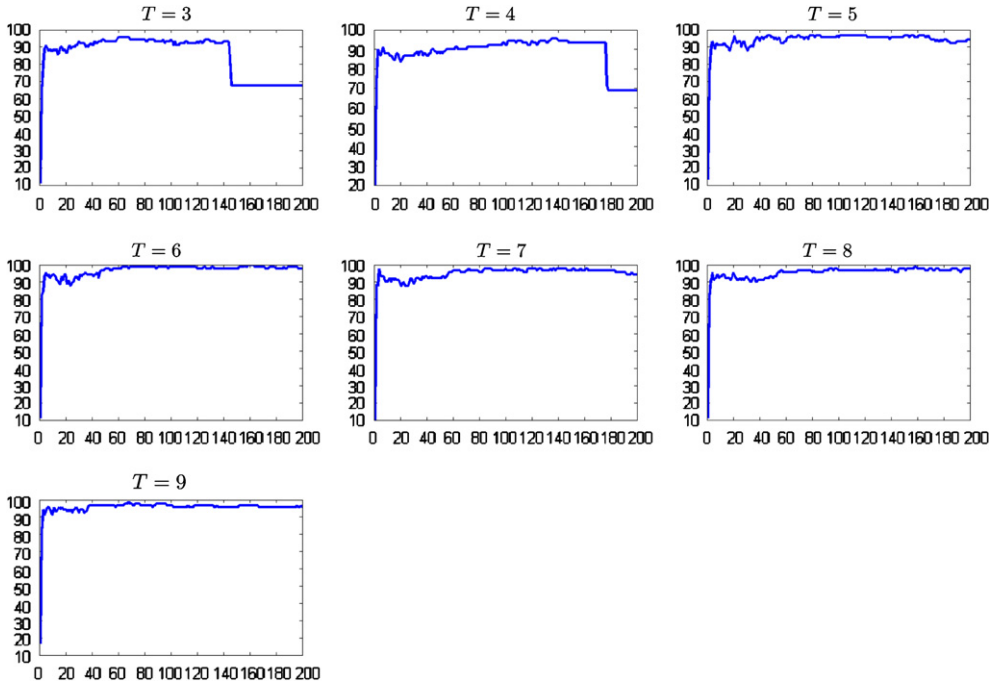

Fig. 11. Recognition accuracy versus dimension (l), for different values of T for MHD as the distance metric.

When using MHD, query sequences are not required to have the same length. So we also investigate using the test sequences without warping. From Fig. 13, we have accuracy near 100%, which shows the efficiency of our method even when the query sequence is not warped. The maximum recognition rate without warping the test sequence has dropped slightly to 98.89%. This is due to the lower distances between warped intra-class sequences. We expect the minimum distance between warped intra-class point sets in Eq. (2) to be the distance between similar postures which is small. However, this is not necessarily true without warping.

5.1.2. Comparison with different dimension reduction methods

In this section we compare PDE with well-known dimension reduction methods including PCA, LDA and supervised LPP (SLPP). In Table 1, we show the recognition accuracy using different dimension reduction methods compared to PDE and also the average times needed for training and test. Only one period of each sequence is used in the experiments. After embedding into action space, the sequences are compared using MHD, since they have different lengths. The recognition accuracy for PDE is the most, since using PDE the distance between embedded intra-class sequences is the least while the inter-class points are as far as possible. PCA has the second highest accuracy because of the small training set of Weizmann database, since PCA performs better in small training sets [24]. SLPP outperforms LDA. This is probably because LPP is more capable in learning the nonlinear structures of action manifolds. The training time for computing PCA is lower than LDA and SLPP, since PCA is solving a simple eigenvalue problem, but LDA and SLPP involve solving a generalized eigenvalue problem with the same number of samples. Computation of PDE has the least computational time. Although PDE requires solving a generalized eigenvalue problem, but the number of samples used for PDE is much less than other methods. Note that solving a generalized eigenvalue problem is of cubic-time complexity with respect to the number of samples [31]. For instance in Weizmann dataset, if we leave the sequence of run performed by the first person aside for test and train with the remaining sequences, for $T=6$, the number of samples (frames) for PDE is 540 while the number of samples for other methods is 2010, which is an enormous saving in time and complexity. Considering the time needed for warping (10.11 s) and aligning (0.18 s), the total time needed for training our method (11.7 s) is still lower than LDA and SLPP. Also considering the average time needed for warping each sequence (0.11 s), our method needs the least time for query classification. This faster and more effective computation is a great advantage especially when time and memory are critical. If speed is critical, we can choose a smaller T without compromising much on the accuracy.

In order to show how discriminant each method is, the visualization of data points in the action space is shown in Fig. 14, where the 3D subspace regarding the first three main components is illustrated. The points with the same color belong to the same action. Note that here we are studying the distribution of data points and not the sequences. From Fig. 14, PDE appears to have better clustering effect compared to other methods. PCA is the least discriminative one since it ignores the class labels.

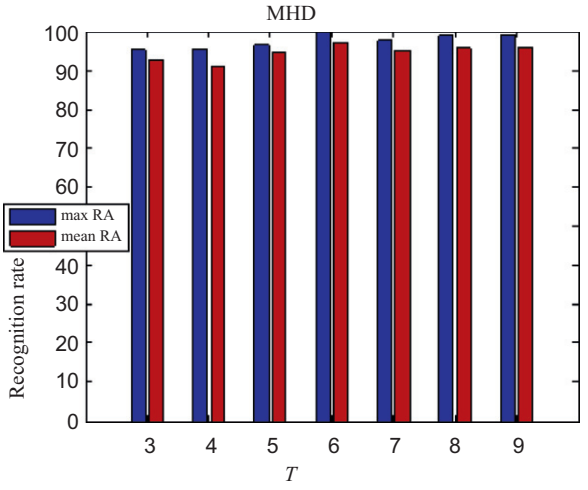

Fig. 12. Studying the effect of T: comparing the maximum and mean of recognition rate in the stable interval from the plots in Fig. 11 for different values of T.

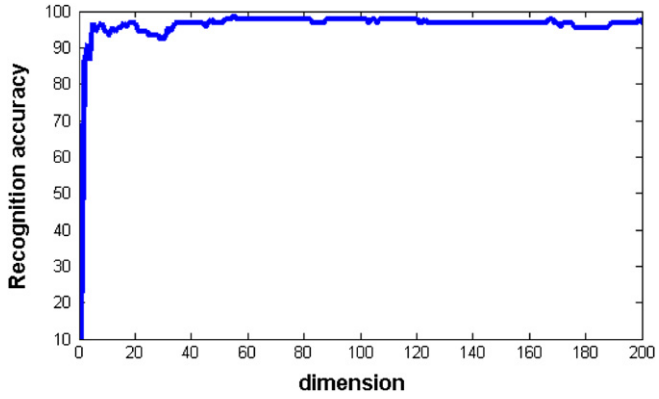

Fig. 13. Recognition accuracy versus dimension, using the test sequences without warping for $T=6$ and MHD as distance.

Table 1
Comparison of different dimension reduction methods. The number in parenthesis shows the optimum dimension. $T=6$ is used for PDE. The experiment is done using a single action cycle.

Dimension Reduction Method	PCA	LDA	SLPP	**PDE**
Recognition accuracy	95.56% (16)	91.11% (9)	93.33% (8)	**100%** (156)
Mean train time (s)	05.94	41.41	40.39	**01.41** (**11.7 with preprocessing**)
Mean test time (s)	00.26	00.26	00.26	**00.04** (**0.15 with preprocessing**)

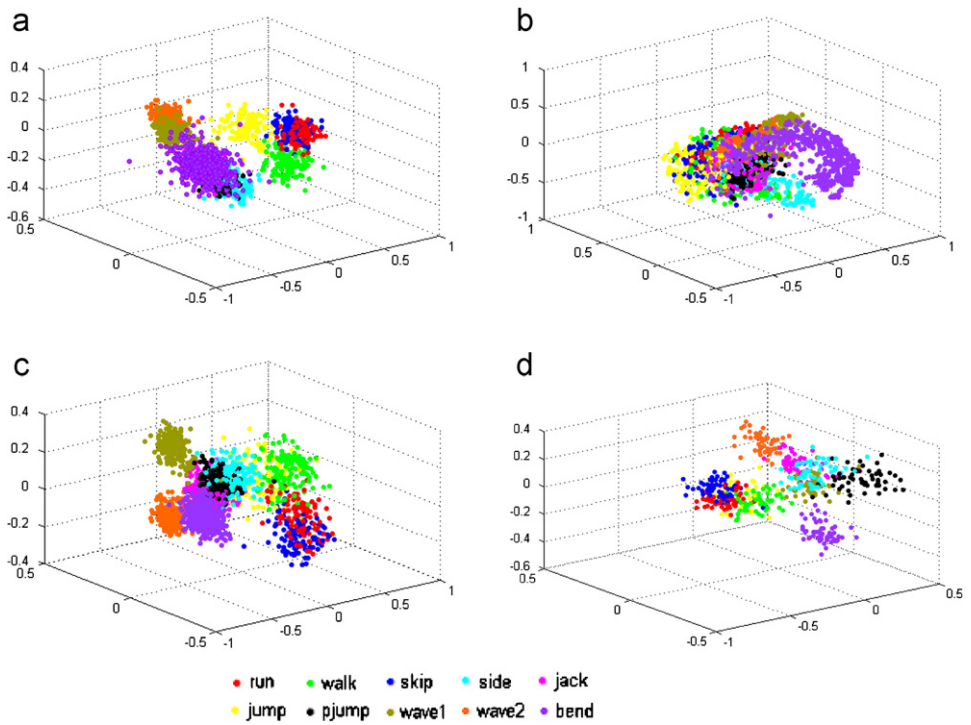

Fig. 14. 3D visualization of action points: (a) LDA, (b) PCA, (c) SLPP and (d) PDE.

The data points embedded by SLPP seem to be more discriminant than those projected by LDA due to power of LPP to find the nonlinear structure of action manifolds.

5.1.3. Comparison with recent results on Weizmann dataset

In Table 2, we compare PDE with some of the recent results reported on the Weizmann database. From the table, we see that our method obtains 100% recognition accuracy. Other methods with the similar accuracy are the manifold learning method of Wang and Suter [9] and those of Lin et al. [32], Schindler and Van Gool [33] and Fathi and Mori [34] that use complex features. The features used in Lin et al. [32] and Schindler and Van Gool [33] are based on information from both shape and motion, while our approach is based on simple shape features. Lin et al. [32] learn an action prototype tree in the joint shape and motion space. Schindler and Van Gool [33] extract features from shape and motion and compare them separately with learned samples. Finally the similarities are concatenated into a single vector which is classified by a bank of linear classifiers. The method of Fathi et al. [34] is based on motion features built from optical flow information and created by a variant of Adaboost. It involves the expensive and sensitive computation of optical flow. The method of Wang and Suter [9] has an accuracy equal to ours, however, the processing times are significantly longer than our method. They use SLPP for learning the action space in a framework similar to ours. But they use one period more than ours,

Table 2
Comparison with recent results on Weizmann dataset.

Method	Accuracy (%)
PDE	**100**
Wang and Suter [9], Lin et al. [32], Schindler and Van Gool [33], Fathi and Mori [34]	100
Wu et al. [11]	98.9
Zhong and Stevens [35]	98.6
Wang and Suter [10]	97.8
Zhang et al. [36]	92.9
Ali et al. [37]	92.6
Jia and Yeung [8]	90.9
Scovanner et al. [38]	84.2
Niebles and Fei-Fei [39]	72.8

i.e. totally two action cycles for each training sequence. This way, there is more information in the training set, which possibly increases the recognition rate. So the recognition rate they achieve is more than the accuracy for SLPP in Table 1, i.e. 93.33%. Accordingly their training time is higher than the time in Table 1, i.e. 40.39 s, which is considerably longer than ours (1.54 s). The test time (0.26 s) is also much higher than PDE (0.02 s). This clearly shows that our method is more efficient in finding the underlying action space.

We have compared our method with different approaches including methods using manifold learning. Methods using manifold learning and dimension reduction techniques [9,11,10,8] have already been reviewed in Section 2. Among the other methods, Zhong and Stevens [35] compute a 3D spatio-temporal volume of motion energy. Local motion descriptors are extracted from the computed volume and compared with the learned feature set in order to encode the action. Zhang et al. [36] represent action videos as motion history images and extract local features from them. Distribution of these local features over relative locations is captured by a histogram similar to shape context. Eventually each action is modeled as a 3D descriptor. Extracted point trajectories are used as features in the method of Ali et al. [37]. These features are classified based on chaotic invariants. Scovanner et al. [38] have extended the SIFT descriptors to form a 3D descriptor. In their method, videos are represented in a bag-of-words framework. Furthermore, Neibles et al. [39] use spatio-temporal features in a hierarchical framework. All these methods have recognition accuracies lower than PDE, which verifies the efficiency of our method in recognizing actions.

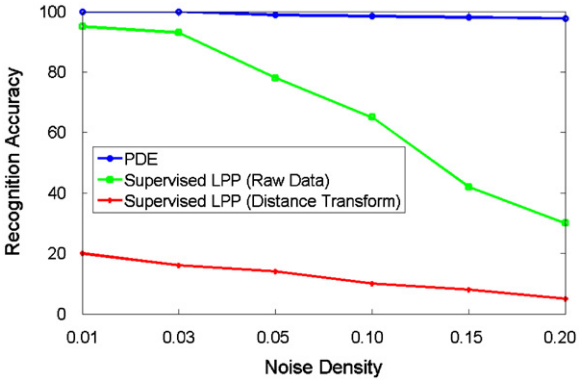

Fig. 15. Results of robustness to noise.

5.1.4. Robustness test

In this section we study the robustness of our method with respect to additive noise, deformations and change in viewpoint.

Robustness to noise: The silhouettes that we use for the experiments are almost free of noise. To check robustness to noise, we add various amounts of synthetic noise to silhouette images to simulate corrupted silhouettes. Since the silhouette images are binary, salt and pepper noise is added. In this experiment the percentage of the affected pixels in the image is shown by noise density. We repeat this experiment for SLPP with *original silhouettes* (used for PDE) as well as *distance transformed silhouettes*. The distance transform is used in [9] to compensate for variation of appearance among different persons. The results are shown in Fig. 15. The proposed method is clearly robust to noise, but the other methods are not. When noise is added, the distance between intra-class sequences might become more than the distance between

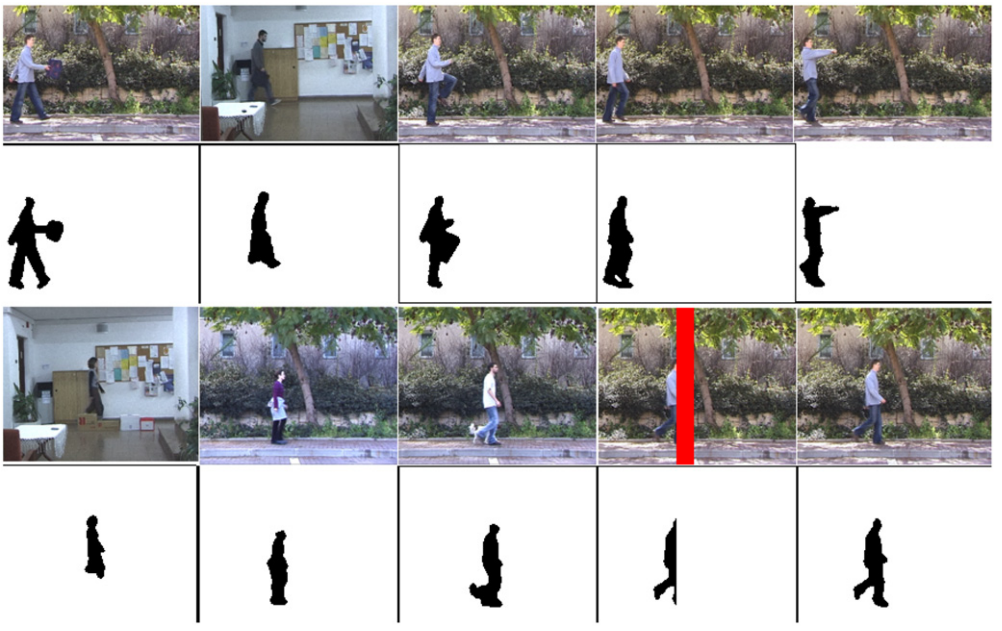

Fig. 16. Sample images of Weizmann's robustness database for deformations [12]. From left to right and from top to bottom: swinging bag, carrying briefcase, knees up, limping man, sleepwalking, occluded legs, walking in a skirt, walking with a dog, presence of a pole, normal walk, respectively.

Table 3
Results of deformation robustness test.

Test sequences	Conditions	Classification result	
		Best match	**Other actions among the 10 best matches**
Swinging bag	Rigid deformation	walk	run, side
Carrying briefcase	Rigid deformation	walk	skip,run
knees up	Walking style	**side**	pjump, **walk**, skip
Limping man	Walking style	walk	side
Sleepwalking	Walking style	**skip**	wave2, jack, side, **walk**, run
Occluded legs	Partial occlusion	walk	pjump, skip, jump
Walking in a skirt	Clothes	walk	side, jump
Walking with a dog	Non-rigid deformation	**run**	**walk**, skip
Presence of a pole	Occlusion	walk	side, run
Normal walk	Background	walk	side, run

Fig. 17. Sample images of Weizmann's robustness database for viewpoint [12]. From left to right and from top to bottom: 0, 5, 10, 15, 20, 25, 30, 35, 40, 45, respectively.

inter-class sequences. The proposed method which guarantees the least distance between intra-class sequences will therefore minimize the classification error. Note that noise affects the distance transformed image more than the original silhouette, because the influence of noise is greatly increased when using distance transform.

Robustness to general deformations: The robustness of the proposed method to some challenging factors such as rigid and non-rigid deformations, variation in clothes and motion styles and also occlusions is investigated in this section. For this purpose we use Weizmann's robustness database for deformations which contains 10 instances of walking with general deformations [12]. Some example images and the corresponding masks are shown in Fig. 16. Each of these test sequences is compared with all the 90 actions in the Weizmann's database to find the best match. Here we do not segment the action cycles. So for the query video, the whole sequence is used with MHD as the distance measure. The point-set matching characteristics of MHD handle the different time durations and aligning. The results are shown in Table 3. Except for three sequences (*knees up*, *sleepwalking and walking with a dog*), all other test sequences are correctly classified as *walk*. This is similar to [9,10] in the number of misclassified sequences. The three misclassified sequences by our method are different with the normal walk in the sense of style and non-rigid deformation. Even for these three sequences, walk is among the 10 best matches. This shows that our method has relatively low sensitivity to changes in clothes, motion style and also rigid and non-rigid deformations and occlusion. In [9,10] *walking with a bag* which is a sort of rigid deformation is being misclassified, instead of *knees up*.

Table 4
Results of viewpoint robustness test.

Test sequence (angle)	Classification result	
	Best match	**Other actions among the 10 best matches**
00	walk	run, skip
05	walk	skip, side, run
10	walk	skip, side
15	walk	skip, side
20	walk	side, skip, pjump
25	walk	pjump, side, skip
30	**pjump**	side, **walk**
35	**pjump**	side, jump
40	**pjump**	side, wave1, bend
45	**pjump**	side

Robustness to change in viewpoint: Changing the view angle causes wide variations in motion and shape of subject and therefore result in error. In this section we study robustness of our method to variations in viewpoint. We use the Weizmann robustness dataset for viewpoints [12]. Samples images of this dataset are illustrated in Fig. 17. Here, also we use the whole sequence for the query with MHD as the distance measure. The results are shown in Table 4. From the table, our method can tolerate up to 30°

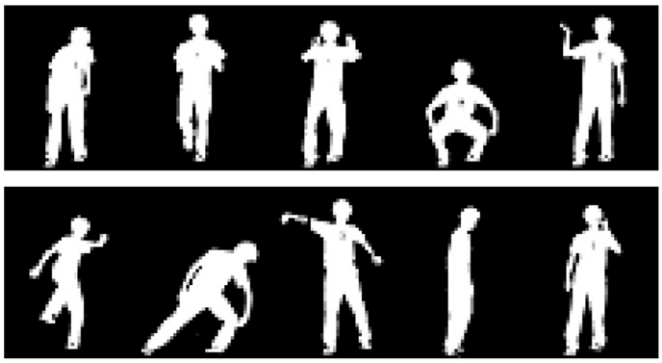

Fig. 18. Examples of silhouettes of Maryland database [13]. From left to right: pick up object, jog in place, push, squash, wave, kick, bend to the side, throw, turn around, and talk on cell phone, respectively.

Table 5
Comparison of different dimension reduction methods. The number in parenthesis shows the optimum dimension. $T=6$ is used for PDE. The experiment is done using a single action cycle.

Dimension Reduction Method	PCA	LDA	SLPP	**PDE**
Recognition accuracy	100% (21)	99% (9)	99% (8)	**100%** (156)
Mean train time (s)	14.98	93.30	97.86	**01.86** (**20.4 with preprocessing**)
Mean test time (s)	01.75	01.75	01.75	**00.04** (**0.22 with preprocessing**)

change in viewpoint which is considerable. From 30° to 45°, the walk action is wrongly classified as *pjump* probably since we have less horizontal movements as well as pjump. Our method is not intrinsically designed for viewpoint change but still it manages to handle large variations in viewpoint angle which is promising.

5.2. Experiments on Maryland database

This dataset [13] comprises 10 different actions performed by one person. There are 10 different instances for each action, resulting in 100 action sequences in this dataset. The instances differ on the temporal rate of execution but there are also slightly different motion styles. The actions are pick up object, jog in place, push, squat, wave, kick, bend to the side, throw, turn around and talk on cell phone. Two cameras with 45° difference in viewpoint have captured these actions. We use the videos from the frontal view. Examples of silhouettes are shown in Fig. 18. The dataset is divided into 10 sets, each containing one instance of all actions. Each time we leave one set out for the test and train using the remaining nine sets. The final result is the average of the 10 runs. We achieved a 100% recognition rate for this database.

The comparison between PDE and other popular dimension reduction techniques is illustrated in Table 5. This experiment is performed using a single action cycle. From Table 5 our method has the best accuracy. PCA is also better than LDA and SLPP, due to small training sample. Similar to Table 1 our method has the best classification time. Also the time needed for training is much lower than LDA and SLPP even considering the preprocessing steps.

5.3. Experiments on KTH database

The KTH dataset is one of the largest and most challenging datasets for human action recognition. This video database contains six types of human actions (walking, jogging, running, boxing, hand waving and hand clapping) performed several times by 25 subjects in four different scenarios: S1:outdoors, S2:outdoors with scale variation, S3:outdoors with different clothes and S4:indoors with lighting variations [2]. Extracting silhouettes in this dataset is quite difficult because of the presence of shadows, severe jitter, lighting variations, camera movement and zoom. As mentioned in [7,40] silhouettes are not very useful in the KTH dataset because of the difficulty in extracting them. This becomes clearer when we note that most of the works on this dataset including [2,4,6,41–43] are based on local features extracted directly from the raw video, avoiding the difficulties in foreground extraction. Furthermore most of the literature dealing with silhouettes for action recognition does not report results on the KTH dataset. In [7], silhouettes have been extracted from a few of the videos of KTH from which a reliable foreground can be extracted, i.e. 36 sequences of walk and 32 sequences of run for a total of 68 sequences out of 2950 video clips. They have only performed a cross dataset test using the extracted video clips as the test samples and the Weizmann dataset for training.

Despite the stated unsuitability of silhouette-based methods for the KTH dataset, we have experimented it with for those actions for which a reasonably clean edge image representing the silhouette can be obtained. These actions are the *in-place* actions in the dataset, viz., boxing, hand clapping and hand waving. We manually draw a bounding box to contain the subject and assume that the position of the bounding box remains the same

Fig. 19. Examples of computed edge maps for in-place actions of KTH dataset.

Table 6
Comparison with other methods on KTH dataset for the in-place actions.

Method	Accuracy (%)
Lin et al. (State of the art) [32]	97.00
PDE	**92.60**
Gilbert et al. [44]	89.67
Nowozin et al. [45]	89.00
Rapantzikos et al. [43]	86.67
Dollar et al. [4]	85.00
Schuldt et al. [2]	77.07
Ke et al. [5]	72.23
Li et al. [46]	65.33

throughout the action in all the frames (except for S2 in which the position and size of bounding box changes[1]). We apply the canny edge detector inside the bounding box that results in a silhouette of the subject. It is these edges that are now embedded in the low-dimensional space. All windowed frames are normalized to 80×48 pixels. Examples of computed edge maps for in-place actions are illustrated in Fig. 19. We see that these edge maps, resulting from manual bounding boxes, are sometimes corrupted and noisy. Table 6 compares the performance of our method with other approaches for the three in-place actions. Clearly our method outperforms every method reported except the state of the art of Lin et al. [32]. Thus, we believe that if we can get reasonably clean edge maps that serve as silhouettes for all the actions in the KTH dataset, the proposed method could achieve a high recognition rate. This is proposed to be part of the future work. In this work, our objective has been to introduce a new faster embedding method for silhouettes.

6. Conclusion

We have proposed a novel embedding method for action recognition that gives the most discriminant embedding based on SCD as the distance measure between sequences of silhouettes. Actions are modeled by sequences of key poses chosen equidistantly during one action period. The poses are embedded into the learned subspace and compared in the projected space by either SCD or MHD. Several experiments are carried out on three popular datasets to demonstrate the efficiency and power of the proposed embedding. In addition to obtaining results comparable to state of the art on all datasets, our method is outperforming other common dimension reduction methods in both the accuracy and time. Moreover, the method is verified to be robust to additive noise and tolerant to occlusions and various deformations. Also it is view-invariant up to promising extents.

[1] In fact, S2 is only the scale variation of S1, so the normalized foregrounds of S2 will be similar to those for S1.

References

[1] A.A. Efros, A.C. Berg, G. Mori, J. Malik, Recognizing action at a distance, in: IEEE Proceedings of the International Conference on Computer Vision, 2003.
[2] C. Schuldt, I. Laptev, B. Caputo, Recognizing human actions: a local SVM approach, in: IEEE Proceedings of the International Conference of Pattern Recognition, 2004.
[3] Y. Yacoob, M.J. Black, Parameterized modeling and recognition of activities, Comput. Vis. Image Understand. 73 (1999) 232–247.
[4] P. Dollar, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, in: IEEE Proceedings of the International Workshop on Performance Evaluation of Tracking and Surveillance, 2005.
[5] Y. Ke, R. Sukthankar, M. Hebert, Efficient visual event detection using volumetric features, in: IEEE Proceedings of the International Conference on Computer Vision, 2005.
[6] J.C. Niebles, H. Wang, L. Fei-fei, Unsupervised learning of human action categories using spatial-temporal words, in: Proceedings of British Machine Vision Conference, 2006.
[7] W. Li, Z. Zhang, Z. Liu, Expandable data-driven graphical modeling of human actions based on salient postures, IEEE Trans. Circuits Syst. Video Technol. 18 (2008) 1499–1510.
[8] K. Jia, D. Yeung, Human action recognition using local spatio-temporal discriminant embedding, in: IEEE Proceedings of the International Conference on Computer Vision and Pattern Recognition, 2008.
[9] L. Wang, D. Suter, Learning and matching of dynamic shape manifolds for human action recognition, IEEE Trans. Image Process. 16 (2007) 1646–1661.
[10] L. Wang, D. Suter, Recognizing human activities from silhouettes: motion subspace and factorial discriminative graphical model, in: IEEE Proceedings of the International Conference on Computer Vision and Pattern Recognition, 2007.
[11] X. Wu, W. Liang, Y. Jia, Incremental discriminative-analysis of canonical correlations for action recognition, in: IEEE Proceedings of the International Conference on Computer Vision, 2009.
[12] M. Blank, L. Gorelick, E. Shechtman, M. Irani, R. Basri, Actions as space-time shapes, in: IEEE Proceedings of the International Conference on Computer Vision, 2005.
[13] A. Veeraraghavan, R. Chellappa, A. Roy-Chowdhury, The function space of an activity, in: IEEE Proceedings of the International Conference on Computer Vision and Pattern Recognition, 2006.
[14] A. Levin, D. Lischinski, Y. Weiss, A closed-form solution to natural image matting, IEEE Trans. Pattern Anal. Mach. Intell. 30 (2008) 228–242.
[15] J. Wang, P. Bhat, R.A. Colburn, M. Agrawala, M.F. Cohen, Interactive video cutout, in: ACM SIGGRAPH, 2005.
[16] E.S.L. Gastal, M.M. Oliveira, Shared sampling for real-time alpha matting, Proceedings of Eurographics 29 (2010) 575–584.
[17] A. Elgammal, C. Su Lee, Inferring 3D body pose from silhouettes using activity manifold learning, in: IEEE Proceedings of the International Conference on Computer Vision and Pattern Recognition, 2004.
[18] C. Sminchisescu, A. Jepson, Generative modeling for continuous non-linearly embedded visual inference, in: Proceedings of the International Conference on Machine Learning, 2004.
[19] R. Cutler, L. Davis, Robust real-time periodic motion detection, analysis, and applications, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 781–796.
[20] P. Turaga, R. Chellappa, V.S. Subrahmanian, O. Udrea, Machine recognition of human activities: a survey, IEEE Trans. Circuits Syst. Video Technol. 18 (2008) 1473–1488.
[21] L. Wang, W. Hu, T. Tan, Recent developments in human motion analysis, Pattern Recognition 36 (2003) 585–601.
[22] I.T. Jolliffe, Principal Component Analysis, Springer-Verlag, New York, 1986.
[23] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces vs. fisherfaces: recognition using class specific linear projection, IEEE Trans. Pattern Anal. Mach. Intell. 19 (1997) 711–720.

[24] A. Martinez, A. Kak, PCA versus LDA, IEEE Trans. Pattern Anal. Mach. Intell. 23 (2001) 228–233.
[25] X. He, P. Niyogi, Locality preserving projections, in: Proceedings of the International Conference on Advances of Neural Information Processing Systems, 2003.
[26] H. Hotelling, Relations between two sets of variates, Biometrika 28 (1936) 321–372.
[27] T.K. Kim, J. Kitter, R. Cipolla, Discriminative learning and recognition of image set classes using canonical correlations, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2007).
[28] S. Nayak, S. Sarkar, B. Loeding, Distribution-based dimensionality reduction applied to articulated motion recognition, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2009) 795–810.
[29] M. Torki, A. Elgammal, Putting local features on a manifold, in: IEEE Proceedings of the International Conference on Computer Vision and Pattern Recognition, 2010.
[30] E. Billauer, Peakdet: peak detection using Matlab. 〈http://www.billauer.co.il/peakdet.html〉, 2008.
[31] G.W. Stewart, Matrix Algorithms Volume II: Eigensystems, SIAM, 2001.
[32] Z. Lin, Z. Jiang, L.S. Davis, Recognizing actions by shape-motion prototype trees, in: IEEE Proceedings of the International Conference on Computer Vision, 2009.
[33] K. Schindler, L. Van Gool, Action snippets: how many frames does human action recognition require?, in: IEEE Proceedings of the International Conference on Computer Vision and Pattern Recognition, 2008
[34] A. Fathi, G. Mori, Action recognition by learning mid-level motion features, in: IEEE Proceedings of the International Conference on Computer Vision and Pattern Recognition, 2008.
[35] Y. Zhong, M. Stevens, Action recognition in spatiotemporal volume, in: IEEE Proceedings of the International Conference on Computer Vision and Pattern Recognition, 2010.
[36] Z. Zhang, Y. Hu, S. Chan, L. Chia, Motion context: a new representation for human action recognition, in: Proceedings of the European Conference on Computer Vision, 2008.
[37] S. Ali, A. Basharat, M. Shah, Chaotic invariants for human action recognition, in: IEEE Proceedings of the International Conference on Computer Vision, 2007.
[38] P. Scovanner, S. Ali, M. Shah, A 3-dimensional sift descriptor and its application to action recognition, in: Proceedings of the ACM International Conference on Multimedia, 2007.
[39] J. Niebles, L. Fei-Fei, A hierarchical model of shape and appearance for human action classification, in: IEEE Proceedings of the International Conference on Computer Vision and Pattern Recognition, 2007.
[40] K. Guo, P. Ishwar, J. Konrad, Action recognition in video by sparse representation on covariance manifolds of silhouette tunnels, in: Proceedings of the International Conference on Pattern Recognition, 2010.
[41] A. Kovashka, K. Grauman, Learning a hierarchy of discriminative space-time neighborhood features for human action recognition, in: IEEE Proceedings of the International Conference on Computer Vision and Pattern Recognition, 2010.
[42] J. Yin, Y. Meng, Human activity recognition in video using a hierarchical probabilistic latent model, in: IEEE Proceedings of the International Conference on Computer Vision and Pattern Recognition, 2010.
[43] K. Rapantzikos, Y. Avrithis, S. Kollias, Dense saliency-based spatiotemporal feature points for action recognition, in: IEEE Proceedings of the International Conference on Computer Vision and Pattern Recognition, 2009.
[44] A. Gilbert, J. Illingworth, R. Bowden, Scale invariant action recognition using compound features mined from dense spatio-temporal corners, in: Proceedings of the European Conference on Computer Vision, 2008.
[45] S. Nowozin, G. Bakir, K. Tsuda, Discriminative subsequence mining for action classification, in: IEEE Proceedings of the International Conference on Computer Vision, 2007.
[46] Z. Li, Y. Fu, T. Huang, S. Yan, Real-time human action recognition by luminance field trajectory analysis, in: Proceedings of the ACM International Conference on Multimedia, 2008.